RVM (Ruby Verison Management) FTW!

Composed by Hemanth.HM <hmadhavarao@sonicwall.com>

RVM is a command line tool which allows us to easily install, manage and work with multiple ruby environments from interpreters to sets of gems.

RVM helps us to maintain and manage multiple versions of ruby and their gems very easily.
One can also have specific ruby versions and local gems for each particular project directory but adding .rvmrc!

This documents explains the basics of RVM, along with installation and most common use cases.

Prerequisites :

RVM uses the following standard GNU tools (some are built into bash):
bash (>= 3.2), awk, sed, grep, which, ls, cp, tar, curl, gunzip, bunzip2

RVM also requires the following libraries (for installing '-head' rubies).
git and subversion

For Ubuntu :

sudo apt-get install build-essential openssl libreadline6 libreadline6-dev curl git-core zlib1g zlib1g-dev libssl-dev libyaml-dev libsqlite3-0 libsqlite3-dev sqlite3 libxml2-dev libxslt-dev autoconf libc6-dev ncurses-dev automake libtool bison subversion

For Fedora and CentOS :

yum install -y gcc-c++ patch readline readline-devel zlib zlib-devel libyaml-devel libffi-devel openssl-devel make bzip2 autoconf automake libtool bison iconv-devel

NOTE: For centos >= 5.4 iconv-devel is provided by glibc

Installation :

Installing RVM is a simple curl call!

#Install RVM :

```
$ bash -s stable < <(curl -s https://raw.github.com/wayneeseguin/rvm/master/binscripts/rvm-installer)
```

Reload your shell environment:

```
$ source ~/.bash_profile
```

Install ruby :

```
$ rvm install 1.9.3
```

Configuration :

RVM has a configuration directory located in $rvm_config_path which for a typical installation as user to their $HOME directory will be located in ~/.rvm/config/

Defaults

RVM has a set of defaults recorded in the file **$rvm_config_path/db** This file is replaced each time you upgrade RVM so do not edit it.

Overriding Default Settings

In order to override RVM's default settings, place the appropriate key=value entry into the file **$rvm_config_path/user** RVM will then use these settings instead of RVM's defaults.

RVM Usage :

Installing new versions   : rvm install <ruby_version>
Using/Switching versions : rvm use <ruby_version>

List of available MRI Rubies :

1.8.6[-p420], 1.8.6-head, 1.8.7[-p357], 1.8.7-head ,1.9.1-p378, 1.9.1[-p431], 1.9.1-head
1.9.2-p180, 1.9.2[-p290], 1.9.2-head, 1.9.3-preview, 1.9.3-rc1, 1.9.3[-p0], 1.9.3-head, ruby-head

Example usage :

```
$ rvm install 1.9.2,1.8.7               # Install different versions.
$ rvm use 1.9.2                         # To use 1.9.2 and switch between versions.
$ rvm use 1.9.2 --default              # To make 1.9.2 default.
$ rvm list                              # To get the list of installed interpreters.
```

Using specific versions and gem sets of different projects/dirs :

By creating .rvmrc files with the required versions in the respective project dirs will indicate rvm to pickup that version and it's specific local gems.

Example :

```
mkdir -p ~/projects/smb ~/projects/eseries  ~/projects/exp

cd ~/projects/smb
rvm --rvmrc --create 1.8.7@smb

cd ~/projects/eseries
rvm --rvmrc --create 1.9.2@eseries

cd ~/projects/projectc/
rvm --rvmrc --create 1.9.3@exp
```

Now if one cd's to the respective projects, rvm will prompt for the first time if it's safe to use the .rvmrc and if YES, the version specified in the .rvmrc will be default to that dir.

Upgrading :

```
$ rvm get stable        #To upgrade to the most stable version

$ rvm get latest        # Installs the latest released version.

$ rvm get head          # Upgrading to the latest repository source version (the most bugfixes)
```

Concluding RVM : Cut Rubies with ease!

“Not only does RVM make installing multiple ruby interpreters / runtimes easy and consistent, it provides features such as gemsets that aren't typically supported out of the box on most ruby installs.

RVM also lets you use different rubies in a manner that wont mess with your existing ruby install (unless you tell it to) as well as letting you run multiple different rubies in separate terminals concurrently!”